Claims

We claim:

1. A computer-implemented method for adaptively filtering messages routed across a network, the messages rejected based on a set of rejection rules, the method comprising:

extracting, from the rejected messages, attributes that triggered a rejection rule;

for each attribute, determining a frequency with which the messages having the attribute were rejected by the rejection rule; and

generating an exception rule to the rejection rule which rejected the messages with the attribute, responsive to the frequency of the attribute exceeding a threshold.

2. The method of claim 1, further comprising:

allowing a rejected message to pass according to the exception rule.

3. The method of claim 2, wherein allowing a rejected message to pass further comprises:

identifying an attribute of the rejected message that triggered a rejection rule;

for the triggered rejection rule, identifying an exception rule that matches the attribute; and

applying the exception rule to the rejected message to determine whether to allow the message.

4. The method of claim 1, wherein the attribute is one of a message component, a value, a data type, and a length.

5. The method of claim 1, wherein the frequency is a weighted count of occurrences of the attribute.

6. The method of claim 1, wherein the frequency is a direct count of occurrences of the attribute.

7. The method of claim 1, wherein the rejected messages are URL requests, each URL request having at least one URL component, the method further comprises:

maintaining a frequency for each instance of a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected by a rule;

selecting a URL component according to a set of constraints; and

generating an exception rule for the selected URL component and its descendants.

8. The method of claim 7, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected URL component.

9. The method of claim 7, wherein the set of constraints is selecting a URL component with a frequency exceeding a threshold and having no children with a frequency above the threshold.

10. The method of claim 7, wherein the set of constraints is selecting a URL component with the frequency exceeding a threshold.

11. The method of claim 7, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and the number of occurrences with which descendants of the URL component were rejected by the rule.

12. The method of claim 1, wherein the messages are URL requests, each URL request having at least one URL component, and the method further comprises:

storing rejected URLs in a trie structure, wherein each node in the trie structure is associated with a URL component;

maintaining a frequency for each node associated with a URL component, wherein the frequency is a function of a number of occurrences with which a URL component associated with a node and its descendants were rejected with a rule;

selecting a node in the trie structure according to a set of constraints; and

generating an exception rule for the selected node and its descendants.

13. The method of claim 12, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected node.

14. The method of claim 12, wherein the set of constraints is selecting a node with the frequency exceeding a threshold.

15. The method of claim 12, wherein the set of constraints is selecting a node with a frequency exceeding a threshold and having no children with a frequency above the threshold.

16. The method of claim 1, wherein the threshold is a product of a total number of messages over a time interval and a percentage of the messages that should be allowed to pass.

17. A system for adaptively filtering messages routed across a network, the messages rejected based on a set of rules, the system comprising:

a learning engine, for extracting, from the messages, attributes of the messages that triggered a rejection rule, for determining, for each attribute, a frequency with which the messages having the attribute were rejected by the rejection rule, and for generating an exception

rule to the rejection rule which rejected the messages with the attribute, responsive to the frequency exceeding a threshold; and

a filter, for applying the exception rule to rejected messages to determine whether to allow the rejected messages.

18. The system of claim 17, wherein the filter is further adapted to:

identify an attribute of the rejected message that triggered a rejection rule;

for the triggered rejection rule, identify an exception rule that matches the attribute; and

apply the exception rule to a rejected message to determine whether to allow the message.

19. The system of claim 17, wherein the attribute is one of a value, data type, and length.

20. The system of claim 17, wherein the frequency is a weighted count of occurrences of the attribute.

21. The system of claim 17, wherein the frequency is a direct count of occurrences of the attribute.

22. The system of claim 17, wherein the rejected messages are URL requests, each URL request having at least one URL component, and the learning engine further adapted to:

maintain a frequency for each instance of a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected with a rule;

select a URL component according to a set of constraints; and

generate an exception rule for the selected URL component and its descendants.

23. The system of claim 17, wherein the set of constraints is selecting a URL component with the frequency exceeding a threshold and having no children with a frequency above the threshold.

24. The system of claim 17, wherein the set of constraints is selecting a URL component with the frequency exceeding a threshold.

25. The system of claim 22, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and a number of occurrences with which descendants of the URL component were rejected by the rule

26. The system of claim 22, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected URL component.

27. The system of claim 17, wherein the messages are URL requests, each URL request having at least one URL component, and the learning engine is further adapted to:

store rejected URLs in a trie structure, wherein each node in the trie structure is associated with a URL component;

maintain a frequency for each node associated with a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected with a rule;

select a node according to a set of constraints; and

generate an exception rule for the selected node and its descendants.

28. The system of claim 17, wherein the set of constraints is selecting a node with the frequency exceeding a threshold.

29. The system of claim 17, wherein the set of constraints is selecting a node with a frequency exceeding a threshold and having no children with a frequency above the threshold.

30. The system of claim 27, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected node.

31. The system of claim 17, wherein the threshold is a product of a total number of messages over a time interval and a percentage of the messages that should be allowed to pass.

32. A computer program product comprising:

a computer-readable medium having computer program code embodied therein for adaptively filtering messages routed across a network, the messages rejected based on a set of rejection rules, the computer program code adapted to:

extract, from the rejected messages, attributes of the messages that triggered a rejection rule;

for each attribute, determine a frequency with which the messages having the attribute were rejected by the rejection rule; and

generate an exception rule to the rejection rule which rejected the messages with the attribute, responsive to the frequency exceeding a threshold.

33. The computer program product of claim 32, wherein the computer program code is further adapted to:

allow a rejected message to pass according to the exception rule.

34. The computer program product of claim 33, wherein the computer program code is further adapted to:

identify an attribute of the rejected message that triggered a rejection rule;

for the triggered rejection rule, identify an exception rule that matches the attribute of the rejected message; and

apply the exception rule to the rejected message to determine whether to allow the message.

35. The computer program product of claim 32, wherein the attribute is one of a value, data type, and length.

36. The computer program product of claim 32, wherein the frequency is a weighted count of the occurrences of the attribute.

37. The computer program product of claim 32, wherein the frequency is a direct count of the occurrences of the attribute.

38. The computer program product of claim 32, wherein the rejected messages are URL requests, each URL request having at least one URL component, wherein the computer program code is further adapted to:

maintain a frequency for each instance of a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected by a rule; select a URL component with the frequency exceeding a threshold and having no children with a frequency above the threshold; and

generate an exception rule for the selected URL component and its descendants.

39. The computer program product of claim 32, wherein the computer program code is further adapted to generate the exception rule by inferencing a scalar data type of the descendants of the selected URL component.

40. The computer program product of claim 38, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and the number of occurrences with which descendants of the URL component were rejected by the rule.

41. The computer program product of claim 32, wherein the rejected messages are URL requests, each URL request having at least one URL component, wherein the computer program code is further adapted to:

store rejected URLs in a trie structure, wherein each node in the trie structure is associated with a URL component;

maintain a frequency for each node associated with a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected with a rule; select a node with the frequency exceeding a threshold; and

generate an exception rule for the selected node and its descendants.

42. The computer program product of claim 32, wherein the computer program code is further adapted to generate the exception rule by inferencing a scalar data type of the descendants of the selected node.

43. The computer program product of claim 32, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and a number of occurrences with which descendants of the URL component were rejected with the rule.

44. The computer program product of claim 32, wherein the computer program code is further adapted to determine the threshold as a product of a total number of messages over a time interval and a percentage of the messages that should be allowed to pass.